
Basic Communication manager

— Prepared by: Mohab Ahmed —

Table of Contents

- Project description
- Layered Architecture
- Modules/Drivers
- Project flowchart
- Sequence diagram

Project Description

There are 2 MCUs : MCU_1 and MCU_2

MCU_1 will send [BCM operating] to MCU_2

When MCU_1 finish sending, LED_0 in MCU_1 will be toggled

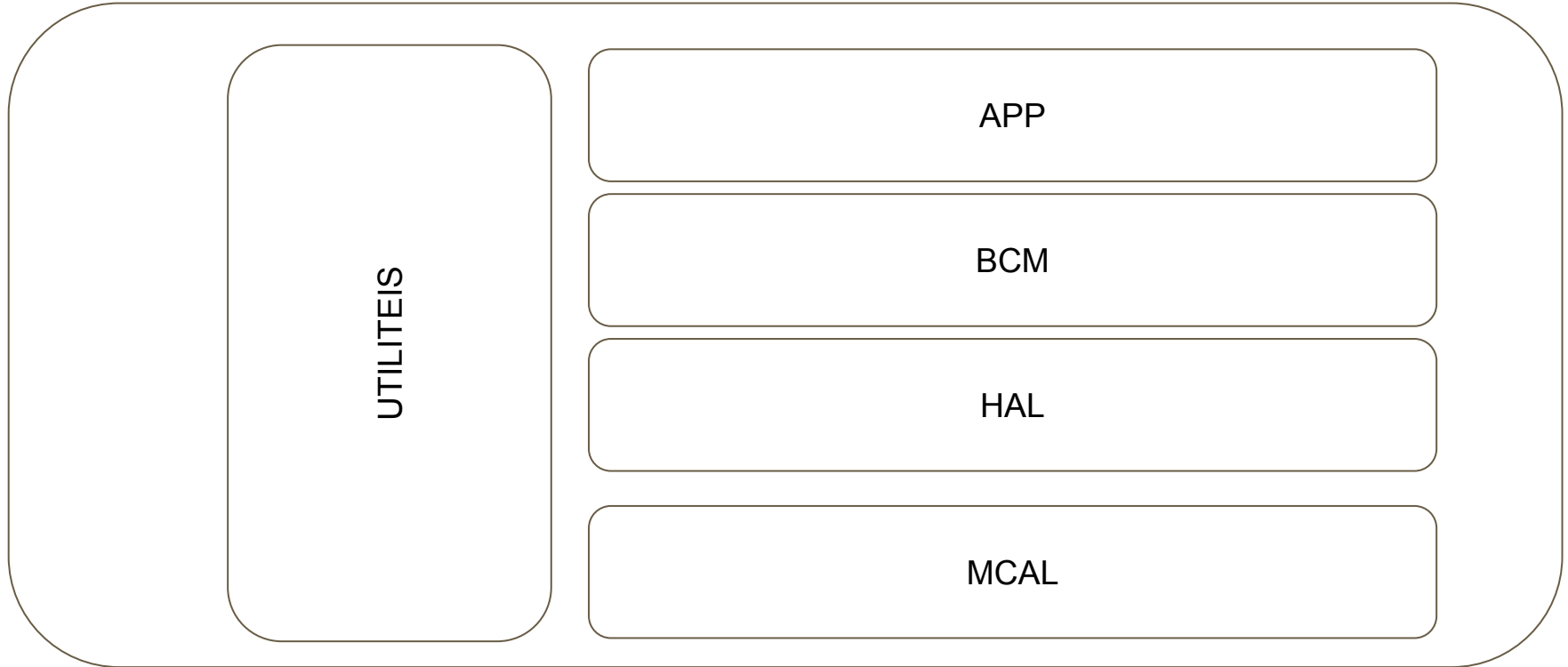
When MCU_2 finish receiving, LED_1 in MCU_2 will be toggled

MCU_2 will confirm with sending [confirm BCM operating] to MCU_1

When MCU_2 finish sending, LED_0 in MCU_2 will be toggled

When MCU_1 finish receiving, LED_1 in MCU_1 will be toggled

Layered architecture



Modules/drivers

Bit_manipulation, registers,
standard types

APP

bcm

Led

Dio, Timer_0, UART

Modules Description

- DIO: It is Digital Input/Output module and it is responsible for dealing with the input and output digital pins.
- Timer_0: It is the timer module which deals with timers in the microcontroller and enables us to make delays.
- UART: Universal Asynchronous Receiver transmitter is a communication protocol which manages the communication between devices.
- LED: It is the module that controls the LEDs through the Microcontroller.
- BCM: Manages the relation between the program and different communication channels.
- APP: It is the module containing the application itself.

Drivers Documentation

- **DIO**

1. `PinDirection_t DIO_setpindir(uint8_t u8_a_portid , uint8_t u8_a_pinid , uint8_t u8_a_pindir);`

Description: sets the direction of the GPIO to either Input or Output.

Arguments: takes the port and pin numbers and pin direction either Input or output.

Return: Return Error statement from `VALID_DIRECTION` or `NOT_VALID_DIRECTION`.

Continue Drivers Documentation

2. `PinValue_t DIO_setpinvalue(uint8_t u8_a_portid , uint8_t u8_a_pinid ,
uint8_t u8_a_pinval);`

Description: This function sets a specified pin to high or low.

Arguments: takes the port and pin numbers and the value of the Pin.

Return: returns the error statement from `VALID_VALUE` or `NOT_VALID_VALUE`.

Continue Drivers Documentation

3. `PinRead_t DIO_readpin(uint8_t u8_a_portid , uint8_t u8_a_pinid ,
uint8_t* u8_a_val);`

Description: This function reads the state of the pin in case of input either high or low.

Arguments: takes the port and pin numbers and a variable to put the value inside.

Return: returns the error statement from `VALID_READ` or `NOT_VALID_READ`.

Continue Drivers Documentation

- **Timer_0**

1. `TMR0_init_error TMR0_init(void);`

Description: sets the direction of the GPIO to either Input or Output.

Arguments: nothing to take.

Return: Return Error statement from `VALID_INIT` or `NOT_VALID_INIT`.

Continue Drivers Documentation

2. TMR0_start_error TMR0_start(void);

Description: starts the timer to count.

Arguments: nothing to take.

Return: Return Error statement from VALID_START or NOT_VALID_START.

Continue Drivers Documentation

3. TMR0_stop_error TMR0_stop(void);

Description: stops the timer to count.

Arguments: nothing to take.

Return: Return Error statement from VALID_STOP or NOT_VALID_STOP.

Continue Drivers Documentation

4. `TMR0_delay_error TMR0_delayms(uint32_t u32_a_delayms);`

Description: delays the system for a period of time.

Arguments: takes the time to be delayed for in milliseconds.

Return: Return Error statement from `VALID_DELAY` or `NOT_VALID_DELAY`.

Continue Drivers Documentation

- **UART**

1. `uart_errorstatus UART_init(void);`

Description: initializes the UART communication protocol with certain settings.

Arguments: nothing to take.

Return: Return Error statement from UART_VALID or UART_NOT_VALID.

Continue Drivers Documentation

2. `uart_errorstatus UART_sendchar(uint8_t ua_a_data);`

Description: send a character through UART.

Arguments: takes the character to be send.

Return: Return Error statement from UART_VALID or UART_NOT_VALID.

Continue Drivers Documentation

3. `uart_errorstatus UART_sendstr(uint8_t * ua_a_string);`

Description: send a group of characters through UART.

Arguments: takes the string to be send.

Return: Return Error statement from UART_VALID or UART_NOT_VALID.

Continue Drivers Documentation

4. `uart_errorstatus UART_receivechar(uint8_t *u8_a_recdata);`

Description: receive a character through UART.

Arguments: takes a pointer to save the character in.

Return: Return Error statement from UART_VALID or UART_NOT_VALID.

Continue Drivers Documentation

5. `uart_errorstatus UART_receivestr(uint8_t * u8_a_recstring);`

Description: receive a group of characters through UART.

Arguments: takes a pointer to save the character in.

Return: Return Error statement from UART_VALID or UART_NOT_VALID.

Continue Drivers Documentation

- **LED**

1. `uint8_t LED_init(uint8_t Port_number, uint8_t Pin_number);`

Description: initializes LED port and pin.

Arguments: takes the port and pin numbers.

Return: Return Error statement.

Continue Drivers Documentation

2. `uint8_t LED_ON(uint8_t Port_number, uint8_t Pin_number);`

Description: turn the Led pin high.

Arguments: takes the port number and the pin number.

Return: Return Error statement.

Continue Drivers Documentation

3. `uint8_t LED_OFF(uint8_t Port_number, uint8_t Pin_number);`

Description: turn the Led pin off.

Arguments: takes the port number and the pin number.

Return: Return Error statement.

Continue Drivers Documentation

4. `uint8_t LED_toggle(uint8_t Port_number, uint8_t Pin_number);`

Description: toggle the Led pin.

Arguments: takes the port number and the pin number.

Return: Return Error statement.

Continue Drivers Documentation

- **BCM**

1. `enu_system_status_t bcm_init(str_bcm_meduim_t* ptr_str_bcm_meduim);`

Description: initializes BCM with the required communication type and settings.

Arguments: takes a pointer with of the BCM medium with the settings.

Return: Return Error statement with the expected error or the valid case.

Continue Drivers Documentation

2. `enu_system_status_t bcm_deinit(str_bcm_meduim_t* ptr_str_bcm_meduim);`

Description: deinitialize the BCM.

Arguments: takes pointer which describes the BCM medium.

Return: Return Error statement.

Continue Drivers Documentation

3. `uint8_t bcm_send(uint8_t ar_data);`

Description: sends the required character.

Arguments: takes the character to be send.

Return: Return Error statement.

Continue Drivers Documentation

4. `uint8_t bcm_send_n(const uint8_t* ar_data, int n);`

Description: sends the required group of characters.

Arguments: takes a pointer to a string.

Return: Return Error statement.

Continue Drivers Documentation

5. `uint8_t bcm_receive(uint8_t *ar_data);`

Description: receives a character .

Arguments: takes a pointer to store the character in.

Return: Return Error statement.

Continue Drivers Documentation

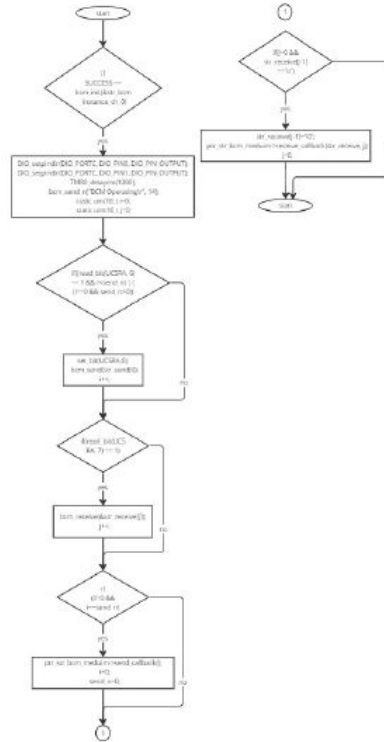
6. `int bcm_receive_n(uint8_t* ar_data,int n);`

Description: receives a group of characters .

Arguments: takes a pointer to store the string in.

Return: Returns number of characters.

Project flowchart



sequence diagram

